PHILOSOPHICAL TRANSACTIONS A

rsta.royalsocietypublishing.org

Heterotic computing: exploiting hybrid computational devices

Viv Kendon[1], Angelika Sebald[2,4] and Susan Stepney[3,4]

[1]Department of Physics, Durham University, Durham DH1 3LE, UK
[2]Department of Chemistry, [3]Department of Computer Science, and [4]York Centre for Complex Systems Analysis, University of York, York YO10 5DD, UK

VK, 0000-0002-6551-3056; SS, 0000-0003-3146-5401

Current computational theory deals almost exclusively with single models: classical, neural, analogue, quantum, etc. In practice, researchers use ad hoc combinations, realizing only recently that they can be fundamentally more powerful than the individual parts. A Theo Murphy meeting brought together theorists and practitioners of various types of computing, to engage in combining the individual strengths to produce powerful new heterotic devices. 'Heterotic computing' is defined as a combination of two or more computational systems such that they provide an advantage over either substrate used separately. This post-meeting collection of articles provides a wide-ranging survey of the state of the art in diverse computational paradigms, together with reflections on their future combination into powerful and practical applications.

Introduction

Cite this article: Kendon V, Sebald A, Stepney S. 2015 Heterotic computing: exploiting hybrid computational devices. *Phil. Trans. R. Soc. A* **373**: 20150091. http://dx.doi.org/10.1098/rsta.2015.0091

Accepted: 5 May 2015

One contribution of 13 to a Theo Murphy meeting issue 'Heterotic computing: exploiting hybrid computational devices'.

Subject Areas:
hybrid computing, theory of computing

Keywords:
heterotic computing, unconventional computing, hybrid computing

Author for correspondence:
Viv Kendon
e-mail: viv.kendon@durham.ac.uk

1. Overview

Practical computation has long used different types of computational components in combination. Everyday examples include the graphics co-processing unit that has cooperated with the central processing unit in desktop and laptop computers for two decades, and the GPS chips included in most mobile phones and digital cameras.

Our vision for hybrid computational systems [1–3] is far broader than this, however, encompassing novel substrates: from the exquisitely controlled quantum systems prototyping a new breed of faster computation based on quantum logic, to the biological and chemical computational experiments in laboratories around

THE ROYAL SOCIETY PUBLISHING

the world. One thing these systems have in common is the use of more conventional classical computers to provide controls for the experiments. On closer inspection, these conventional computers often contribute a crucial part of the computational power of the exotic devices. We use the term 'heterotic' to mean the composition of two or more potentially widely differing kinds of physical computational substrates to produce a computer that has certain computational advantages over an individual system alone.

Most theory of computation deals with single computational paradigms, so we lack a formal basis to analyse these compositions of very different systems. This Theo Murphy meeting issue provides a step on the route to developing such a basis. It brings together a collection of papers dealing with a wide range of unconventional computational substrates. The papers address experimental and theoretical systems, individual and composed systems. Understanding the computational capabilities of individual substrates, and understanding how they can be composed to perform novel kinds of computation, is essential for the next generation of special-purpose, embedded, post-Moore's law devices.

The issue starts with Kendon *et al.* [4], in which we extend our definition of physical computing [5] to include multiple substrates composed, in series or in parallel, into hybrid computational systems. We then flesh out what we mean by 'heterotic' computing through a collection of examples from quantum to biological. Among the recurring features of our examples, we note the importance of including transduction between these substrates in any analysis, because extra computation can be hiding in the transduction process. Indeed, in a paper in this issue, Nehaniv *et al.* [6] explicitly model transduction as 'transformers' connecting automata into networks, and their 'transformers' are themselves a special type of automaton. Another aspect common to many of our examples is the incomplete understanding of such systems due to their hybrid nature being little studied as such. Hence, we also lay out the requirements for future work to address this.

2. Biological computing

Nature has provided us with a rich diversity of exquisitely honed biological systems that can support computation in many forms. Taking inspiration from nature, our next three papers exemplify the huge range of opportunities biological computation provides, and also lay out some of the challenges in the way of realizing this potential.

Despite nature providing rich systems that we can exploit as the basis for computation, such systems still need to be engineered into devices to perform our desired computations. Amos *et al.* [7] provide an overview of the results from their EU FET BACTOCOM project, addressing this issue. Their aim is to develop a method to design bacterial component computers, through exploiting the exchange of plasmid genetic material through biological conjugation, which can be regarded as a communication protocol. Classical computing is embedded in the design loop, with simulation of various bacterial properties.

One of the more exotic systems used for laboratory-based computational experiments is *slime mould*, described by Adamatzky [8]. Exploiting the behaviour of this living organism by providing food and barriers to growth allows simple computations to be performed, including the lovely image on the journal cover. Place food at the entrance and exit of a maze, and the slime mould will configure itself along the (approximate) shortest path between them. The heterotic nature of this computation consists of the slime mould organism, its container designed to enable the particular computation to be performed, and the image processing required to extract the resultant path. Elementary logic gates using slime mould can be designed. However, designing gates is one thing; persuading the slime mould to compose them into circuits is another: the slime mould connects end to end in its structures, and multiple gate sequences turn out not to work as predicted. This illustrates the importance of using an appropriate model for the physical computer: binary logic is not the best fit for slime mould. This also illustrates why it is insufficient to demonstrate that an unconventional substrate can implement logic gates in order to show classical computational universality: it must also be possible to solve the wiring problem, and

implement circuits out of those gates. Slime mould's intrinsic capabilities force us to think more creatively about computation, as this article amply illustrates.

Bacteria and slime moulds are complete organisms, albeit single-celled ones. Biological computation can also exploit parts of organisms, such as neurons. Hughes *et al.* [9] review and discuss a particular form of heterotic computing: neural systems interfacing silicon substrates. Although both systems use electricity in some form to communicate, they have great differences in function and form. The authors describe some specific technologies for combining neurons and silicon, discuss the pros and cons, and draw out the very practical potential applications if the problems of implementing the interface *in vivo* can be overcome.

3. Molecular computing

Biological substrates exhibit a rich diversity, but these will have all of the unpredictable complexities of any biological system. Nature also provides simpler molecular substrates that still have suitably complex computational capabilities. The four papers in this section address various molecular-based computational substrates, both inorganic systems, and biomolecules.

Neural-like behaviour does not necessarily need neurons. Gorecki *et al.* [10] use the Belousov–Zhabotinsky chemical reaction as an analogue of important behaviours of neural systems, namely thresholds and spiking. They note how their systems are heterotic: the main computation is performed in the chemical substrate, but external systems are additionally needed for input, output and reaction rate control. These systems include illumination to control excitability in different regions of the medium, and microfluidics to engineer and control droplet systems. Enclosing the reactants in droplets that allow exchange with the environment via diffusion adds another layer of structure to the system, and paves the way for applications to sensors that can compute with the same substrate as they detect, instead of using a conventional computer chip that needs to be protected from the environment and interfaced with the detector.

Computation can occur not just when a fixed body of material changes state, but also when material (self-)assembles into some desired form. Woods [11] surveys theoretical models of 'tile assembly', small structures joining together according to local rules to form specific two-dimensional patterns; such tiles can be implemented using DNA as a construction material. With judicious design of the tiles (the 'program' defining what can stick to what), a 'universal' tile set can be achieved. Indeed, there is even a single (albeit rather complicated) universal tile. The computational power (which structures can be built) of different systems is defined in terms of *simulation*: which tile sets can assemble the same patterns (maybe at different scales) as other sets. Here the theory is discussed; in a physical implementation, much imaging and processing would be required to observe the assembled patterns.

Jonoska & Seeman [12] also use DNA tiles to perform computation, but in a rather different manner from the previous paper. Above we have pure self-assembly with local assembly rules embodied in the tile designs. Here, the tile system is arranged to execute a two-dimensional cellular automaton (CA) with synchronization (controlling when tiles can bind and unbind) implemented with clocked external light sources. A fixed layer of tiles forms the background CA grid; a set of floating tiles can selectively bind and unbind at grid positions. The grid is designed to signal which tiles should bind and unbind, depending on the current local state of bound tiles, thus implementing the CA logic; the currently bound tiles form the CA pattern. This approach, instead of building up to a programmed structure, results in a dynamic process of tile rearrangement as the system proceeds though its states.

These two DNA-based articles are using DNA as a readily available and highly configurable base system to explore the generic concepts of assembly. The ideas and results are applicable to any substrate capable of sufficiently complex self-assembly, allowing the fields of material self-assembly and computational self-assembly to overlap and interoperate. This also illustrates another basic attribute of physical computation that is easily forgotten with the ubiquity of binary encoding: the fungibility of representation. The same computation can be embedded in different computational substrates and even embedded in different ways in the same substrate [5].

Henson *et al.* [13] describe a system comprising a computer, a robot and a chemical reaction system, linked together in a fully automated experimental system to search for chemical products that exhibit interesting complex spatio-temporal dynamics (motile and dividing droplets). Importantly, the procedure provides reproducible synthetic pathways to such products. The complete integration of the computation with the synthesis, including feedback loops, is crucial to produce an effective exploration of the chemical state space. The chemicals so discovered are themselves potential candidates for an unconventional droplet computation. This substitution of highly programmed search over top-down design will be a feature in the construction of many unconventional devices, particularly while we lack a well-developed design approach for these complex and ill-understood substrates. The tight integration between the computer and the system it is controlling also suggests we are approaching territory in which self-reproduction of computational machines is not tied to replication of the physical system; rather, the computation being represented becomes the connecting factor between generations.

4. Models and theories

Our conventional, classical models of computation were developed for specific purposes. The diversity in novel computational substrates and their behaviours requires new theoretical computational models that fully represent the possibilities offered by the physical substrates. The four papers in this section address a broad range of such theoretical and modelling issues.

Software engineering requires tools and techniques to support the programming process. Heterotic systems will require adaptations of conventional techniques and inventions of new approaches. Stannett & Gheorghe [14] describe how the well-established X-machine testing paradigm can be extended to cover a particular case of heterotic computing, that of a controlling state machine communicating with a second, unconventional, system acting as an 'oracle'. They discuss what further research is needed to extend this approach to full heterotic systems, with continuous and continual time evolutions. Such testing will be essential for deploying novel computational devices in any mission critical (i.e. useful) role.

Taking multi-device computation to the logical limit, Beal & Viroli [15] review the main aspects of field computing, a model where computation is distributed across space and time. Although the authors do not explicitly address heterotic systems, their collective computation of aggregates is an important component of many spatially distributed substrates. The unifying theory they develop draws on ideas from computational physics, and has broad application, from computation in a spatially extended continuous chemical substrate communication via diffusion (e.g. [10]), to a network of conventional devices communicating wirelessly with local neighbours. An overarching framework for programming distributed networks of computational devices provides the tools required for effective and widespread use of such systems.

Building on many years of prior work, Nehaniv *et al.* [6] present a mathematical model based on finite automata for diverse biological systems, that can be extended hierarchically to any level of aggregation of constituent parts. They identify the symmetries and the substructures that lead to interesting behaviour and interpret how control and dependencies flow through the network. As already noted, they explicitly include *transduction* (here called *transformers*) between finite automata when they connect them into networks. Most interesting from a biological perspective is their extension to dynamically changing networks, where constituents can be added or removed as required. These models can handle cell division as naturally as a new digit representing 'tens' appears when we add 7 and 8 to get 15. Among many important applications, this provides a path to modelling computational self-assembly and construction, such as discussed by Woods [11]. The connections to computation are elucidated through the deep role of groups in the mathematical structure of these networks, where 'SNAGs'—simple non-Abelian groups—imply the finite automata are capable of the finite equivalent of universal computation. Such SNAGs are present even in simple biological systems, such as the p53-mdm2 genetic regulatory system that governs a cell's response to radiation damage. The results and insights in this article represent

a significant advance in our understanding of complex systems and will have applications far beyond the examples presented to illustrate the ideas.

Deciding whether biological entities are computing, or are maybe 'universal' in some other biological sense, requires a notion of what it means for a physical system to compute. Horsman *et al.* [5] have recently developed abstraction/representation theory, to formalize when a physical system is indeed computing, rather than merely 'doing its thing'. Horsman [16] summarizes that theory, then shows how it can be applied to multiple substrates in the case of hybrid and heterotic computational systems. This reveals new structure and possibilities in the case of composed systems, and the author uses the approach to distinguish two different forms of composition: of substrate (computational medium) and of representation (essentially, how the results of the computation are observed and interpreted). Substrate composition results in a system that is straightforwardly the sum of its parts, whereas representation composition yields a novel kind of system whose *computational* behaviour cannot readily be decomposed into the separate actions of its substrates. As different physical substrates can support wildly different kinds of representation, this work demonstrates how novel forms of computation, above and beyond single substrate results, can be achieved in heterotic systems.

5. Summary

This issue contains a wide selection of highlights from the vibrant diversity of research in unconventional computation. From laboratory-based experiments with mainstream applications in sight, to foundational theory that deepens our understanding of the nature of computation and computation in nature, this is only 'unconventional' in contrast with our current digital silicon-based devices. Many of these diverse systems and creative ideas are destined for mainstream significance in the future of computation in the twenty-first century. If this issue collectively has one message, it is how vast the possibilities are that are waiting to be explored. We invite you to feast on the articles herein, and to be inspired to generate your own contributions to the flourishing arena of heterotic computing.

Conflict of interests. We declare we have no competing interests.

Funding. S.S. acknowledges partial funding by the EU FP7 FET Coordination Activity TRUCE (Training and Research in Unconventional Computation in Europe), project reference no. 318235. V.K. is funded by EPSRC fellowship EP/L022303/1.

Acknowledgements. We thank the Royal Society for the support of the Theo Murphy international scientific meeting on *Heterotic computing: exploiting hybrid computational devices*, held at the Kavli Royal Society International Centre for the Advancement of Science at Chicheley Hall, 7–8 November 2013, which was the origin of this Theo Murphy meeting issue. We thank all the meeting attendees, and particularly the rapporteurs, for engaging in excellent stimulating open debates. Their thought-provoking ideas helped form this issue. We thank all the authors for their splendid contributions, and the referees for their hard work in ensuring the main articles are all of the highest standard.

References

1. Kendon V, Sebald A, Stepney S, Bechmann M, Hines P, Wagner RC. 2011 Heterotic computing. In *Unconventional computation* (eds CS Calude, J Kari, I Petre, G Rozenberg). Lecture Notes in Computer Science, vol. 6714, pp. 113–124. Berlin, Germany: Springer. (doi:10.1007/978-3-642-21341-0_16).
2. Stepney S, Abramsky S, Bechmann M, Gorecki J, Kendon V, Naughton TJ, Perez-Jimenez MJ, Romero-Campero FJ, Sebald A. 2012 Heterotic computing examples with optics, bacteria, and chemicals. In *Unconventional computation and natural computation* (eds J Durand-Lose, N Jonoska). Lecture Notes in Computer Science, vol. 7445, pp. 198–209. Berlin, Germany: Springer. (doi:10.1007/978-3-642-32894-7_19).
3. Stepney S, Kendon V, Hines P, Sebald A. 2012 A framework for heterotic computing. In *8th Workshop on Quantum Physics and Logic* (*QPL 2011*), *Nijmegen, The Netherlands*. EPTCS, vol. 95, pp. 263–273.

4. Kendon V, Sebald A, Stepney S. 2015 Heterotic computing: past, present and future. *Phil. Trans. R. Soc. A* **373**, 20140225. (doi:10.1098/rsta.2014.0225)
5. Horsman C, Stepney S, Wagner RC, Kendon V. 2014 When does a physical system compute? *Proc. R. Soc. A* **470**, 20140182. (doi:10.1098/rspa.2014.0182)
6. Nehaniv CL, Rhodes J, Egri-Nagy A, Dini P, Rothstein Morris E, Horváth G, Karimi F, Schreckling D, Schilstra MJ. 2015 Symmetry structure in discrete models of biochemical systems: natural subsystems and the weak control hierarchy in a new model of computation driven by interactions. *Phil. Trans. R. Soc. A* **373**, 20140223. (doi:10.1098/rsta.2014.0223)
7. Amos M, Axmann IM, Blüthgen N, de la Cruz F, Jaramillo A, Rodriguez-Paton A, Simmel F. 2015 Bacterial computing with engineered populations. *Phil. Trans. R. Soc. A* **373**, 20140218. (doi:10.1098/rsta.2014.0218)
8. Adamatzky A. 2015 Slime mould processors, logic gates and sensors. *Phil. Trans. R. Soc. A* **373**, 20140216. (doi:10.1098/rsta.2014.0216)
9. Hughes MA, Shipston MJ, Murray AF. 2015 Towards a 'siliconeural computer': technological successes and challenges. *Phil. Trans. R. Soc. A* **373**, 20140217. (doi:10.1098/rsta.2014.0217)
10. Gorecki J, Gizynski K, Guzowski J, Gorecka JN, Garstecki P, Gruenert G, Dittrich P. 2015 Chemical computing with reaction–diffusion processes. *Phil. Trans. R. Soc. A* **373**, 20140219. (doi:10.1098/rsta.2014.0219)
11. Woods D. 2015 Intrinsic universality and the computational power of self-assembly. *Phil. Trans. R. Soc. A* **373**, 20140214. (doi:10.1098/rsta.2014.0214)
12. Jonoska N, Seeman NC. 2015 Molecular ping-pong Game of Life on a two-dimensional DNA origami array. *Phil. Trans. R. Soc. A* **373**, 20140215. (doi:10.1098/rsta.2014.0215)
13. Henson A, Gutierrez JMP, Hinkley T, Tsuda S, Cronin L. 2015 Towards heterotic computing with droplets in a fully automated droplet-maker platform. *Phil. Trans. R. Soc. A* **373**, 20140221. (doi:10.1098/rsta.2014.0221)
14. Stannett M, Gheorghe M. 2015 Integration testing of heterotic systems. *Phil. Trans. R. Soc. A* **373**, 20140222. (doi:10.1098/rsta.2014.0222)
15. Beal J, Viroli M. 2015 Space–time programming. *Phil. Trans. R. Soc. A* **373**, 20140220. (doi:10.1098/rsta.2014.0220)
16. Horsman DC. 2015 Abstraction/Representation Theory for heterotic physical computing. *Phil. Trans. R. Soc. A* **373**, 20140224. (doi:10.1098/rsta.2014.0224)